

Spacecraft Terminal Emulator

ITOS User's Manual

\$Date \$

Copyright 1999-2006, United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code.

This software and documentation are controlled exports and may only be released to U.S. Citizens and appropriate Permanent Residents in the United States. If you have any questions with respect to this constraint contact the GSFC center export administrator, <Thomas.R.Weisz@nasa.gov>.

This product contains software from the Integrated Test and Operations System (ITOS), a satellite ground data system developed at the Goddard Space Flight Center in Greenbelt MD. See <<http://itos.gsfc.nasa.gov/>> or e-mail <itos@itos.gsfc.nasa.gov> for additional information.

You may use this software for any purpose provided you agree to the following terms and conditions:

1. Redistributions of source code must retain the above copyright notice and this list of conditions.
2. Redistributions in binary form must reproduce the above copyright notice and this list of conditions in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:
This product contains software from the Integrated Test and Operations System (ITOS), a satellite ground data system developed at the Goddard Space Flight Center in Greenbelt MD.

This software is provided ‘‘as is’’ without any warranty of any kind, either express, implied, or statutory, including, but not limited to, any warranty that the software will conform to specification, any implied warranties of merchantability, fitness for a particular purpose, and freedom from infringement and any warranty that the documentation will conform to their program or will be error free.

In no event shall NASA be liable for any damages, including, but not limited to, direct, indirect, special or consequential damages, arising out of, resulting from, or in any way connected with this software, whether or not based upon warranty, contract, tort, or otherwise, whether or not injury was sustained by persons or property or otherwise, and whether or not loss was sustained from or arose out of the results of, or use of, their software or services provided hereunder.

Designed for the Triana mission, the Spacecraft Terminal Emulator (ITOS_TERM) is an ITOS related program to process an interactive terminal session to the spacecraft using an xterm window interface to send directives to the spacecraft OS using special spacecraft commands via STOL and receive replies in a specific telemetry packet. Optional parameters allow out put to be written to a file exclusively or in parallel with screen output. ITOS_TERM is started by STOL procs `scterm` or `scfterm` which acquires the current running telemetry stream. For starting a proc, see section "START" in *"ITOS STOL Directives"*. Note, there should be no "default" entry in the 'ctrlsource.dat' file. All double quotes entered in the xterm to be sent to the spacecraft are replaced with 2 double quotes.

- Command line arguments for ITOS_TERM.
- Proc arguments for SCTERM.
- Proc arguments for SCFTERM.

0.1 Command-line arguments for itos_term

```
itos_term -t <txport> -h <host> -p <port> -o <offset> -c <command> -f <file> -v <int>
```

```
-t <txport> Transport over which to get data: 'udp'
'client_tcp', 'server_tcp', 'file'. Defaults
to 'client_tcp' if a <host> is given, to
'file' if a <file> is given, else to 'server_tcp'.
```

```
-h <host> Host to which to connect for data under the
client_tcp transport, and the host from which
to accept data under the udp transport (defaults
to any host). Under udp, class D IP addresses
cause the program to join the given multicast
group. Required for client_tcp transport
```

```
-p <port> Port to which to connect for data under the
client_tcp transport, or the port on which to
listen under the server_tcp transport, or the
port on which to read under the udp transport.
Required for client_tcp transports. For other
transports, defaults to 0 which causes the
system to select the port.
```

```
-o <offset> Offset into packet to start extracting data. The length
of the data is to the end of the packet and can thus be
variable in length.
```

```
-x <type> Defines the datatype packet string (S1 or S21).
```

```
-c <command> Command string to prefix to operator typeins
before sending it to STOL. If <command> is "NOTERM"
then no xterm window is created. This requires "-f"
```

option to be selected otherwise an error occurs and the program terminates.

`-f <file>` The name of a file write output. If omitted, output only goes to the xterm window.

`-v <int>` Set verbosity level `<int>` for debugging. Defaults to 0.

0.2 Proc invocation arguments for `scterm.proc`

`start scterm(operation, station, virtual_channel, appid, offset, type, prefix)`

<code>operation</code>	[begin, end], "begin" open a new xterm session, "end" closes a running xterm session. If a begin is used and an existing xterm of the same parameters is still running then that window is brought to the foreground and this proc exits.
<code>station</code>	[the entry name in 'ctrlsource.dat' used for the acquire of telemetry].
<code>virtual_channel</code>	[0 .. 7], the VC number the return strings comes down, most likely vc0.
<code>appid</code>	[0 .. 4095], the packet number return strings come down.
<code>offset</code>	[number], the offset from the beginning of the packet where the return string starts. The length of the data is to the end of the packet and can thus be variable in length.
<code>type</code>	[S1, S21], S1 strings are received in the natural order, i.e. "ABCD" comes down as "ABCD", S21 strings are received with every other character swapped, i.e. "ABCD" comes down as "BADC".
<code>prefix</code>	[quoted string] containing STOL spacecraft command to be prefixed to the string entered in the xterm window to be sent to the spacecraft]. Example: <code>"/hkshell command="</code> .

Example: `scterm(begin, dynsim, 0, 3, 36, S1, "/ismshell command=")`

starts an xterm window connected to a data stream called "dynsim" that will acquire all VC0 AppID 3 packets and strip out an S1 type string starting at offset 36 till the end of the packet and displayed it in the xterm window. All keyboard entered lines will be appended to the `"/ismshell command="` string and sent to STOL to be executed.

Example: `scterm(end, dynsim, 0, 3, 36, S1, "/ismshell command=")`

terminates the ITOS_TERM program started by the above example. It is best to terminate this way rather than just closing the xterm window so that the running frame_sorter task is terminated as well.

0.3 Proc invocation arguments for scfterm.proc

```
start sfcterm(operation, station, virtual_channel, appid, offset, type, prefix, file)
  operation      [begin, end], "begin" open a new xterm session,
                  "end" closes a running xterm session.  If a begin
                  is used and an existing xterm of the same parameters
                  is still running then that window is brought to the
                  foreground and this proc exits.

  station        [the entry name in 'ctrlsource.dat' used for
                  the acquire of telemetry].

  virtual_channel [0 .. 7], the VC number the return strings comes down,
                  most likely vc0.

  appid          [0 .. 4095], the packet number return strings come down.

  offset         [number], the offset from the beginning of the packet where the
                  return string starts. The length of the data is to the
                  end of the packet and can thus be variable in length.

  type           [S1, S21], S1 strings are received in the natural order,
                  i.e. "ABCD" comes down as "ABCD",
                  S21 strings are received with every other character
                  swapped, i.e. "ABCD" comes down as "BADC".

  prefix         [quoted string] containing STOL spacecraft command
                  to be prefixed to the string entered in the xterm
                  window to be sent to the spacecraft]. Example:
                  "/hkshell command=". If string is set to "NOTERM" (all
                  capitals) then the xterm window is not open and all
                  output is sent to the "file". If file is "" then
                  "NOTERM" is ignored.

  file           [quoted string] containing the fully qualified pathname
                  of a file to write the output. If the string is empty
                  (i.e. "") then no file output is created.
```

Example: `scfterm(begin, dynsim, 0, 3, 36, S1, "/ismshell command=", "")`

starts an xterm window connected to a data stream called "dynsim" that will acquire all VC0 AppID 3 packets and strip out an S1 type string starting at offset 36 till the end of the packet and displayed it in the xterm window. All keyboard entered lines will be appended to the "/ismshell command=" string and sent to STOL to be executed. No output is written to a file so this proc is effectively the same as "scterm".

Example: `scfterm(begin, dynsim, 0, 3, 36, S1, "/ismshell command=", "scterm.out")`
 starts an xterm window exactly the same as the above example but output is simultaneously written to the file "scterm.out".

Example: `scfterm(begin, dynsim, 0, 3, 36, S1, "NOTERM", "scterm.out")`
 starts the itos_term program with the same options as the above examples but no xterm window is created and output is only written to the file "scterm.out".

To stop the above example and any itos_term invocation, start the same proc that started the program but change the operation parameter to "end".

Example: `scfterm(end, dynsim, 0, 3, 36, S1, "NOTERM", "scterm.out")`